Face Recognition Based on Singular valued Decomposition and BackProgagation Neural Network

Thaahirah S. M. Rasied, Othman O. Khalifa and Yuslina Binti Kamarudin

International Islamic University of Malaysia
Faculty of Engineering
khalifa@iiu.edu.my

***Abstract*-This paper presents a developed face recognition system. The method used singular valued decomposition as images feature extractor and back propagation neural networks as its classifier. The back propagation training parameters are varied in order to find the best parameter with the highest performance accuracy. The results from experiments have showed that combinations of both methodologies give good recognition rate and therefore considered as an effective face recognition system.**

1. INTRODUCTION

Automatic Recognition of the human faces has been an area of active research for the past few years. This is due to the fact that the need for its applications to secure access controls, financial transactions etc is growing. Therefore, with its emerged applications through various fields, automatic identification of human faces or generally known as biometric recognition system have a new importance in real world now days. The biometrics recognition technique acts as an efficient method and has wide applications in the areas of information retrieval.

Therefore, it is designed to verify or recognize the identity of a living person on the basis of his/her biometric features such as face, fingerprint, iris or some aspects of behavior such as handwritings. However, it has been stated that face is the most reliable among those features because it is free of many problems [1]. Practically, faces less sensitive to such small injuries and it is very stable compared to other biometric features.

Hence great progress has been made in face recognition in past 20 years. Among the existed techniques that have been proposed are eigen faces [1][4][6], wavelet/elastic matching [2][6], PCA and neural networks [2][5][6]. For almost all these proposed techniques, the success rate of face recognition depends on the solution of two problems: representation and matching [6]. The representation of a pattern can be considered as feature extraction in pattern recognition. In [7], Hong suggested that the algebraic features are stable and valid features in object recognition such as face recognition. He proposed singular value decomposition (SVD) based method which uses the singular values as the feature extractor and had obtained an acceptable recognition rate.

In general, an unsupervised learning approach cannot get a high recognition rate [2]. Under conditions, where we can not acquire a large number of face images for every person, utilizing all available samples is very important. Therefore, the face verification proposed by Yunhong Wang [8], utilizes Radial Basis Function (RBF) Neural Network as it's classifier and SVD as the feature extractor. Good discrimination ability was obtained with an accuracy rate of 92 %.

However, in this paper, we are proposing a face recognition system based on back-propagation neural network classification method, whose image features are the singular values (SV) of face images, where all images are pre-processed in advance. Figure 1 shows the process of the proposed system.

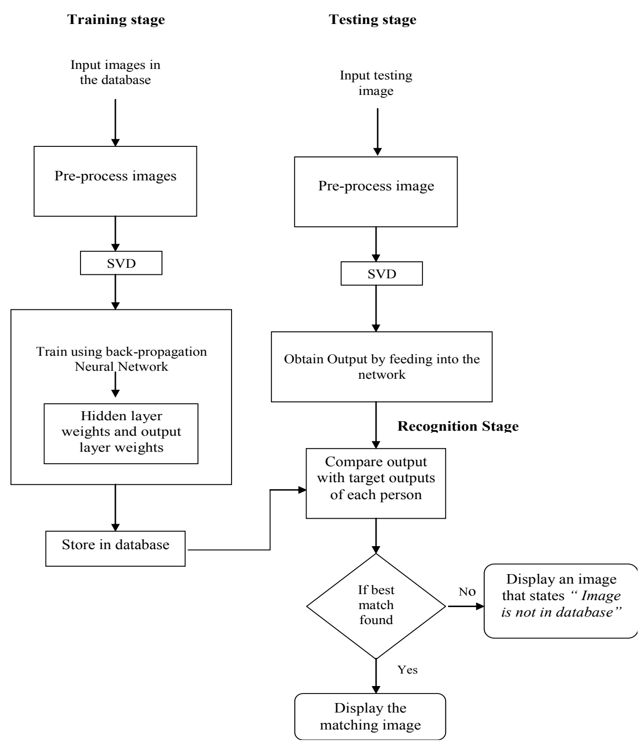

Fig.1: The Block Diagram of the Proposed System

1-4244-0011-2/05/$20.00 ©2005 IEEE.

This system consists of three stages:
Training stage, Testing stage and Recognition stage.
In training stage an artificial intelligence created by man in any recognition system. Machines are programmed to differentiate and identify different images. Therefore, the training stage is very important to train or familiarize the system with all images stored in the library, for it to be able to match them with an unknown image. Firstly, images are pre-processed by using certain technique. Then, singular values as the feature vectors obtained from SVD algorithm to those preprocessed images. Finally, those singular values will be an input vector to the back-propagation neural network, and will be trained using supervised learning. In testing stage, the input image is loaded into the system, pre-processed, feature extracted with obtaining it's singular values. Next, it is feedfoward into all 10 networks that have been built in the system and outputs are obtained. In Recognition stage, the outputs from each network are rounded to either 0 or 1 using certain floating point value. If the rounded output matches or similar to the positive class vector that belongs to that specified network, then, the test image is considered matching with the positive image of the network. The system will try to find matches through all the networks and determine number of matches found. There are three possibilities that could occur: If only one match found, a positive image of the matching network will be displayed as its recognized image. If more than one match found, the system will find the distance of the unrounded output vector with its respective positive target vector. The output vector that has the minimum distance is considered the best match. Hence, a positive image that belongs to that output vector's network will be displayed as its recognized image. If there is no match found, then, the test image is considered as an image that does not match with any images in the database.

II. SINGULAR VALUE DECOMPOSITION

The implementation of Singular Valued Decomposition (SVD) is in fact falls into the face extraction section of face recognition process. Theoretically, singular valued decomposition (SVD) has been defined as one method used to efficiently decrease the amount of data processed. Basically, the basic concept of SVD is to represent an image of size *m x n* as 2D *m x n* matrix. SVD is then applied to this matrix to obtain the ***u***, ***s*** and ***v*** matrices. Where ***s*** matrix is an *n* by 1 matrix known as the diagonal matrix where its values are the singular values of an image. Therefore, in order to determine the next *S* matrix,

$$S = \begin{bmatrix} s_{11} & 0 & \Lambda & 0 \\ 0 & s_{22} & O & 0 \\ M & O & O & 0 \\ 0 & \Lambda & 0 & s_{jj} \end{bmatrix} \qquad \text{Where } s_{11} = \sigma_{11} = \sqrt{\lambda_1}\,.$$

and A *m x n* matrix **A** is determined from an image.

$$u_i = \frac{1}{\sigma_{ii}} A v_i$$

In face recognition face specifically, singular valued decomposition (SVD) method had functioned as a new way for extracting algebraic features from an image. Thus, this method has been introduced and been used in many fields such as data compression, signal processing and pattern analysis. Among the properties which made SVD approach favorable in defining the face recognition process is defined as follows; The SVD of a face image has good stability in which it defined that whenever a small perturbation is added to a face image, a large variance of its singular values (SVs) didn't occur. Previously, it has been stated that singular values represents algebraic properties of an image. Hence, SV features possess algebraic and geometric invariance as an instance [11].

Several theorems related to properties of SVD have been defined by previous researched, in which it described mathematically the characteristic on why this method is favored in face recognition specifically in the face extraction part [2][4][10][11].

In relation with all those determined properties of the SVD approach, this method of face extraction is considered very desirable to be applied into the face recognition. In addition, SVD approach is very much needed in the face recognition field especially when images were taken under different noise and view point conditions.

III. NEURAL NETWORK MODEL

A neural network have been used in the field of image processing, it provides an optimistic result in terms of quality of outcome and ease of implementation. Neural network proved itself to be invaluable in applications where a function based model or parametric approach to information processing are difficult to formulate. The description of neural network can be summarized as a collection of units that are connected in some pattern to allow communication between the units. These units are referred as neurons or nodes generally. The output signals feed to other units along the connections which known as weight. The weights usually excite or inhibit the signal that is being communicated. One of the specialty of neural networks is that the *hidden units* factors. The function of the *hidden units* or *hidden cells* or also called *hidden neurons* is to intervene between the external input and the network output. The network which implemented neural network in it actually has the ability to extract higher-order statistics by adding one or more hidden layers. Hence, this characteristic is particularly valuable when the size of the input layer is large, specifically in the face recognition field [13].
The back-propagation learning algorithm is used to train the weights in the network and updating the weights of a multilayered network which undergoing supervised training. However, in 1974, Werbos [15] has developed a technique for adapting the weights and then in 1986, Rumelhart [16] had

improved it into neural network [12]. This improvement of weight adaptation rule is known as back-propagation.

A. Back-propagation Learning Algorithm

The backpropagation algorithm is a multi-layer network using a weight adjustment based on the sigmoid function, like the delta rule. According to the back-propagation Network (BPN) algorithm, a fully connected feedforward network is assumed. Hence, the activation travels in a direction from input layer to the output layer and the units in one layer are all connected to every unit in the next layer. Basically, back-propagation algorithm consists of two sweeps of the network which are the forward sweep and the backward sweeps. Forward sweep defines the network from the input layer to the output layer, in which it propagates the input vectors through the network to provide outputs at the output layer in the end. During the forward sweep, the weights of the networks are all fixed. The backward sweep hence defines network from the output layer to the input layer, where it is similar to forward sweep except that the error values are propagated back through the network. This is done in order to determine how the weights are to be changed during the training, in which the weights are all adjusted in accordance of an error correction rule where the actual response of the network is subtracted from the target response to produce an error signal [14]. In brief, those error values were passed along the weighted connections in the reverse directions of the forward sweep. The illustration of this algorithm is represented in fig 2, where *i* denoted the input layer, *j* denotes the hidden layer while *k* is the output layer and that this network uses BPN as its' algorithm. x_i and o_k were each defined the input and output vectors of the system.

In fig.2, the shaded hidden units send activation to each output units and thus during backward sweep, this hidden unit will received an error signals from the output units. Basically, the number of processing elements in each layer will vary according to the applications verified.

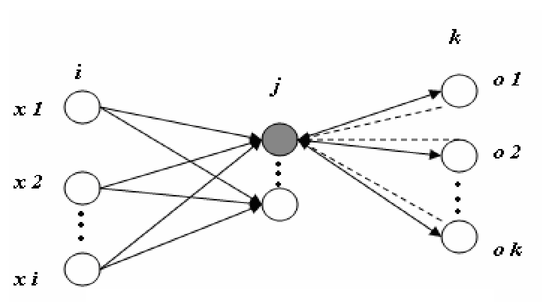

Fig. 2 Typical Structure of

Back-propagation algorithm used supervised learning approach due to the fact that in BPN, the target output vectors are defined earlier in the system. The learning process begins with the presentation of an input pattern to the BPN. In which, the net total input is found using the standard summation of products as defined in Equation 1 below.

$$net_j = \sum_{i=0} x_i w_{ij} \quad \ldots\ldots \quad \ldots\ldots..(1)$$

where; x = input vectors

w = weight vectors

Basically, units have a rule for calculating an output value that will be transmitted to other units, in which this rule is known as an *activation function* and the output value is referred to as the *activation* for the unit. Back-propagation algorithm used sigmoid function as the *activation function*, and represented by Equation 2 below, where $f(net_j)$ denoted the activation function for the hidden layer;

$$f(net_j) = \frac{1}{1+\exp(-net_j)} \quad \ldots\ldots\ldots (2)$$

In order to find the net total output, Equation 3 defined the appropriate formula;

$$net_k = w_k f(net_j) \quad \ldots\ldots\ldots\ldots\ldots\ldots . (3)$$

Then, the output at both hidden layer and output layer are each determined by Equation 4 and 5 as defined below.

$$o_j = f(net_j) \ldots \quad \ldots (4)$$

$$o_k = f(net_k) \ldots\ldots\ldots \quad \ldots\ldots (5)$$

The input pattern is actually propagated through the entire network until the output pattern is produced. Basically, the BPN make used of *generalized delta rule* in order to determine the error for the current pattern contributed by every unit in the network. δ_j denoted the error for all hidden layer units as in Equation 6 and δ_k denoted the error across all the output layer units as in Equation 7.

$$\delta_j = f'(net_j)\sum_{k=1}^{k} \delta_k w_{kj} \quad \ldots\ldots\ldots\ldots\ldots \ldots (6)$$

$$\delta_k = (t_k - o_k) f'(net_k) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Finally, each unit modifies its input connection weights slightly in a direction that reduces its error signal, and then the process is repeated for the next pattern. By applying a learning rate η, the weight change for a unit in hidden layer is determined by;

$$\Delta w_{ij} = \eta \delta_j x_i \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (8)$$

While for output layer, the weight change could be determined from;

$$\Delta w_{kj} = \eta \delta_k f(net_j) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

The very last step in back-propagation is to update the weight values in the system using the following equation.

$$\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n) \quad \ldots\ldots (10)$$

In the Equation 10, the term α is introduced to add in a proportion of the previous weight change, this is due to reduce the likelihood of the weight changes to oscillate. Therefore, the weight change for pattern n + 1 is dependent on the weight change for pattern *n*. Those steps discussed above were repeated for each pattern defined in the systems. The patterns were continually presented to the network, epoch after epoch. This step is conducted until during one epoch, all outputs for each pattern are within tolerance.

IV. IMPLEMTATION AND RESULTS

A set of face images were used each corresponding to variations in facial expression (open/closed eyes, smiling/non-smiling), and facial details (glasses/no glasses). There are 10 different images of 10 individuals from ORL database of Face. All images were taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. The images are grey scale with a resolution of 92 x 112. Experiments were performed with 8 training images and 2 test images for each person. The preprocessing stage is applied in order to distinct different images and the substages are as follows:

1. The original image is eroded or gradually worn off.
2. Then, the original image is used to dilate or make all the features in the image wider.
3. Finally, the pre-processed image is the difference between the dilated image and the eroded image.

Fig. 3 shows an example of the pre-processed image. There are three main experiments are done to analyze the performance of the system presented in this paper.

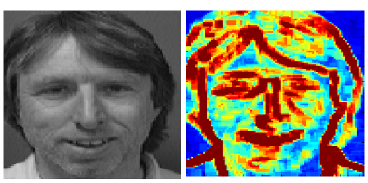

Fig. 3 Sample of image after pre-processing (left: Eroded image, Right: Dilated image)

VI.1 Experiment 1

In the first experiment we studied the recognition performance to determine number hidden cells that contribute to the best accuracy in the system. Hence, recognition rates of both testing and training were computed for 20, 30, 40 and 50 hidden cells after 1000 learning cycles (also known as epoch). In addition to that, the performance of the system is investigated during different learning rates. Learning rates that was used in the training were 0.2, 0.25, 0.3 and 0.35. The results obtained are presented in Fig. 4, 5, 6 and 7. Fig. 4 presents the accuracy of training images after 1000 training cycles. As number of hidden cell and learning rates were varied, the performance of the system varies. The training accuracy was almost 90% during 30 and 40 hidden cells with learning rate equals to 0.2. The system achieved an accuracy of testing equals to 80% after 1000 cycles of training, with a number of hidden cells equals to 30 and it's learning rate was 0.2 as shown in Fig. 5. Analyzing the graphs in Figure 4 and 5, the system was anticipated to have a good performance in terms of training images and testing images accuracy, at hidden cells equals to 30 and 40 which lead us for Experiment 2. However, if the user doesn't choose any of the values; the default values of the parameters will be used to start the training. A graph of Epoches vs Error will be displayed in this interface to show how the errors are reduced during the training.

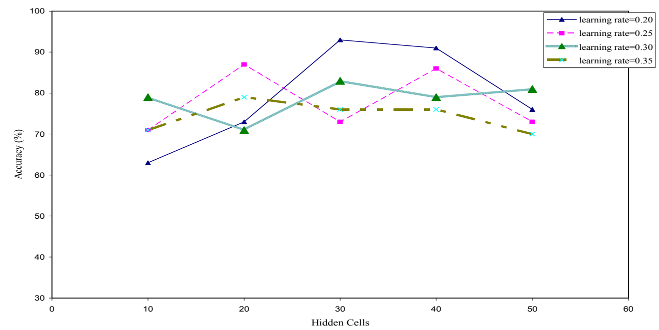

Fig. 4 Graph of Accuracy training images of the system versus number of hidden **cells** at epoch=1000

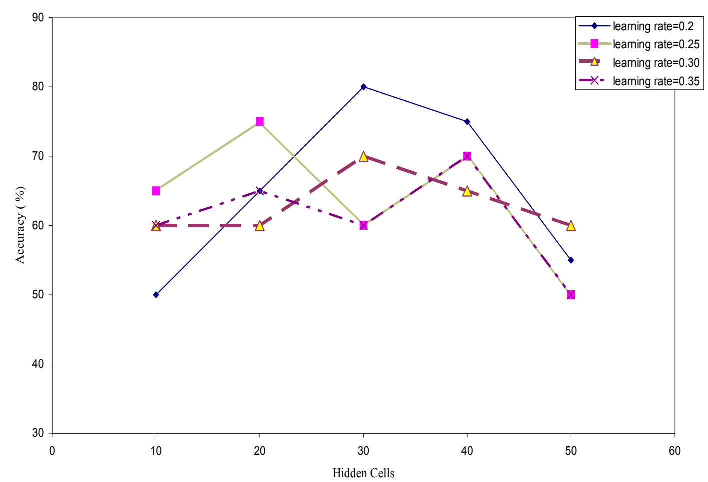

Fig. IV5 Graph of Accuracy testing images of the system versus number of hidden cells at epoch=1000

VI.2 Experiment 2

At this stage, simulations were focused on hidden cells equals to 30 and 40. The learning rates were varied similar as in first stage, and also the number of learning cycles. Recognition rates were computed after 800, 1000, 1500, 2000, 2500, 3000 and 3500 epoch. The main idea of this experiment was, to find a steady accuracy determined from the graphs of a fixed hidden cells, varying learning rates and epoch. As the number of iteration for the training increased, the performance of the system also should be improved until it becomes steady in term of it's recognition rate.

By referring to graph in figure 6, the accuracy of all sets of training images at hidden cells equals to 30 have reached to almost 100% recognition rate after 2500 epoch onwards. As the epoch was increased, the highest testing images accuracy obtained was 80% recognition rate. In addition to that, the testing images recognition rate were constant during simulation for training using learning rate of 0.25 and 0.35 after 2500 epoch onwards. This can be seen clearly in graph in fig. 7. Simulations using hidden cells equals to 40 gave results approximately same as in figure 6 and 7. The results were: at hidden cell equals to 40, training using learning rates equals to 0.25 gives the most stable training and testing images accuracy. At epoch equals to 2500 onwards, the training accuracy had almost reached 100%, while the testing accuracy was 80%.

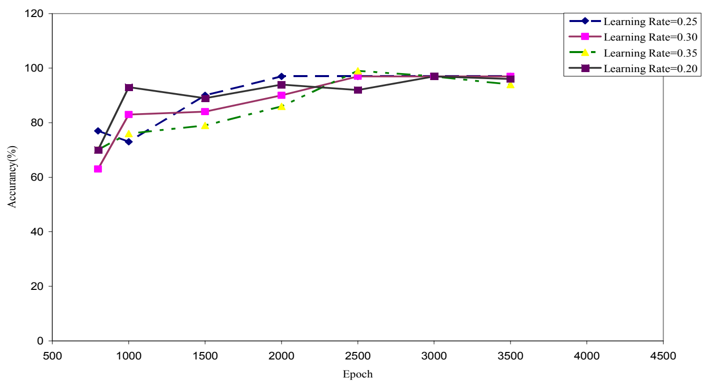

Fig. 6 Graph Accuracy of training images of the system versus number of epoch at Hidden cells=30

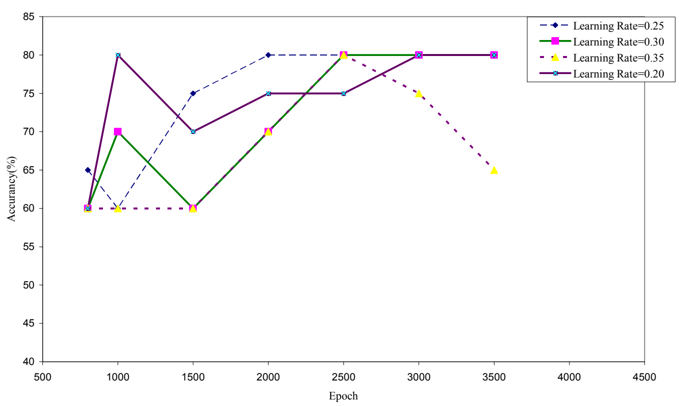

Fig. 7 Graph Accuracy of testing images of the system versus number of epoch at Hidden cells=30

VI.3 Experiment 3

This experiment is to find the performance of the system when the size of the images is varied. Hence, some analyses were done as presented in graph in fig. 8. The original size of image used in this system is 92 x 112. The graph presented shows that the size of images used had been varied with size of 50 x 50 and 30 x 30. The variation of size is very important. All the previous experiment uses the original size. For original size, the size of the input layer cells will be 93 including the bias. However, if the size is reduced to 50 x 50, the input layer cells will 51 and the last variation of size, gives a 31 input layer sizes. Simulations were done on a specific training parameters that had been proved to have one of the best efficiency, which are hidden cells=40, learning rate=0.25. Hence, by analyzing the graphs in figure 8 and 9, the accuracy of recognition of the system highly reduced if the size of images is reduced to a very small size compared to the original size. Even though the variation in the training image recognition were quite high from the original performance of the system, image with size of 50 x 50 still could perform a good recognition on the test images. The performance of images at this size, were approximately similar to the original size of the images.

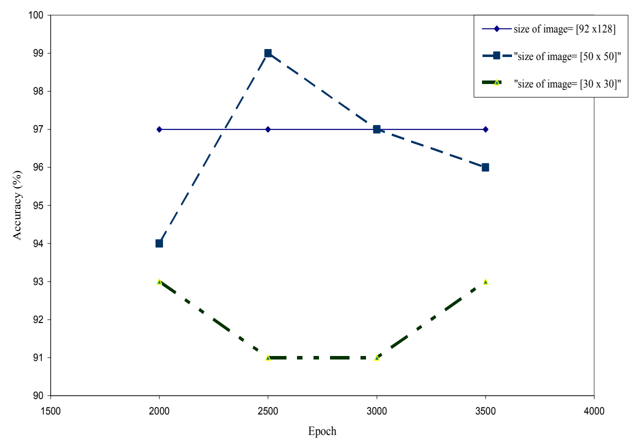

Fig. 8 Graph of training image Accuracy versus epoch at Hidden cell=40, Learning rate=0.25 with variation of size of images

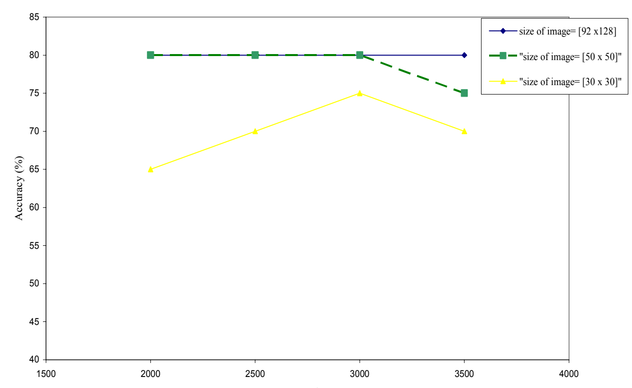

Fig. 9 Graph of testing image Accuracy versus epoch at Hidden cell=40, Learning rate=0.25 with variation of size of images

VII. COMPARISON WITH OTHER FACE RECOGNITION METHODS

Over the past years, a variety of methods for face recognition have been developed, owing to its large scale commercial applications The most popular amongst these techniques is Eigenfaces developed by Turk and Pentland [17], based on the principle of PCA. The basic aim of PCA is to project the input data in the form of face images onto a lower dimensional space in such a manner that the variation between the data is maximized. In other words second-order statistics of the data are de-correlated by this method. The resulting face recognition system is seen to be fairly robust to illumination changes with the added benefit of efficiency resulting from reduction in dimensionality. However, it is seen that higher order statistics are very significant in an application such as face recognition. PCA fails to account for any of the higher order statistics. The proposed method performance was tested using a database of training images in the training phase. The recognition rates were measured on over different images and the performance was observed to be superior to that obtained using PCA.

VIII. CONCLUSION AND DISSCUSSION

This paper presents an algorithm for face recognition by performing singular valued decomposition on the extracted feature of images and then training were done using back propagation neural network where the ORL database of Faces were used. By performing SVD on the extracted images, the size of input layer cells could be reduced by almost 99% compared to the initial size of the image. The performance of the system using this algorithm depends on the parameters like epoch, learning rate and number of hidden cells used in the BPN. Therefore, many experiments were presented in this paper in order to determine at which parameters, the system could give a good performance. The system gave an approximately 90% for training image recognition rate and 77% for testing images at hidden cells equals to 30 and 40; learning rate equals to 0.25 and minimum epoch equals to 2500.

In addition to that, the performance of the system is quite stable and could give an approximately same recognition rate if the size of images trained were reduced by a small factor. By reducing the size of the images, time duration to complete the training could be reduced also. Therefore, this could make the system more efficient

Many future works could be done to test the capability of this system as for example: increasing number of subjects or person used in this system and comparing the system's performance if the SVD is not applied in the system. The system's capabilities to recognize faces in different variations such as illumination, pose, expression, etc can be further explored in order to achieve higher recognition accuracy.

REFERNCES

[1] H Dogu, I.Kaynar and F.T.Yarman Vural., *Face Recognition Using Eigenfaces*, Dept. of Computer Engineering, Middle East technical University, Ankara,Turkey.

[2] A. B William, *A Survey of Face Recognition Algorithms and Testing Results*, National Biometrics Test Center, San Jose State University.

[3] A. Ilker , *Face Recognition Using Eigenfaces*, Institute of Science and Technology, Istanbul Technical University. January, 1996.

[4] C. Leung, *Real Time Face Recognition*, School of Information Technology and Electrical Engineering, University of Queensland. October 2001.

[5] L. Shang-Hung, K. Sun-Yuan and L. Long-Ji, Face Recognition/Detection by Probalistic Decision Based Neural Network, IEE Transaction of Neural Netwroks, Vol. 8, No. 1, January 1997.

[6] Z. Jun , Y. Yong and L. Martian, Face Recognition: Eigenface, Elastic Matching,and Neural Nets, Proceedings of the IEEE, Vol.85, No.9, 1997.

[7] Z. Hong, Algebraic feature extraction of image for recognition, Pattern Recognition, Vol.24, pp211-219, 1991.

[8] W. Yunhong, T. Tieniu and Z. Yong, *Face Verification Based on Singular Value Decomposition and Radial Basis Function Neural Network*, National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, Beijing.

[9] D. R. Hill, B. Kolman, *Modern Matrix Algebra,* Prentice Hall. 2001.

[10] J. Chen, *Image Compression with SVD,* ECS 289K Scientific Computation. 13 December 2000

[11] Y.Wang, T. Tan and Y. Zhu, *Face Verification Based on Singular Value Decomposition and Radial Basis Function Neural Network*, National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, Beijing.

[12] S. Haykin, *Neural Networks, A Comprehensive foundation,* Macmillan College Publishing Company, Inc. 1994.

[13] P. Picton, *Neural Networks*, Second Edition. Palgrave. 2000.

[14] S. Haykin, *Neural Networks, A comprehensive foundation*, Prentice Hall. Second Edition, 1999.

[15] P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, Cambridge, Mass., 1974.

[16] D. Rumelhart,G. R. Hinton and R. Williams, Learning representation by backpropagating errors, Nature 323, 1986, pp. 533-536.

[17] M. Turk and A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3(1), pp. 405-411, 1993.